

Bugs and Errors with Disney WRECK-IT RALPH



Allyssa Loya

Lerner

There are errors in the system! Things are going wrong, but you can make them right. Be a Fix-It Felix, and discover how algorithms and codes can help you solve problems and move characters from one place to another within a gaming universe.



Check It Before You Wreck-It

First, look at the cover.

- Read the title. How do you think a story with this title might teach you something about computer coding?
- What part do you think Ralph plays in the story?

Next, look inside the book.

- Find the Table of Contents and look through the titles. Do you think this book will mostly tell you information? Or do you think it will show you how to use information to do things? Explain.
- Skim through the book. What do you see? Is the book mostly words? Or does it have a lot of diagrams and activities? Explain.
- Find a diagram or picture that looks extra interesting to you. Why do you like it? What can you tell about the book from looking at this one image?



Know the Code!

This book offers coding activities that go along with *Fix It, Felix, Jr.*, *Hero's Duty*, *Sugar Rush*, and the other video games in Litwak's Family Fun Center. These activities feature the vocabulary of computer coding. Much like Ralph, Felix, and Vanellope are part of the *Wreck-It Ralph* video game universe, these words are an important part of the coding universe.

algorithm

decompose

bug

loop

code

resolve



Disney
LEARNING

Disney
Coding
Adventures

Bugs and Errors with Disney WRECK-IT RALPH



Allyssa Loya

Lerner Publications ♦ Minneapolis



Copyright © 2019 Disney Enterprises, Inc. All rights reserved.

No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means—electronic, mechanical, photocopying, recording, or otherwise—without the prior written permission of Lerner Publishing Group, Inc., except for the inclusion of brief quotations in an acknowledged review.

Lerner Publications Company
A division of Lerner Publishing Group, Inc.
241 First Avenue North
Minneapolis, MN 55401 USA

For reading levels and more information, look up this title at www.lernerbooks.com.

Additional graphics provided by Laura Westlund/Independent Picture Service.

Main body text set in Billy Infant Regular 14/20.

Typeface provided by SparkyType.

Library of Congress Cataloging-in-Publication Data

Names: Loya, Allyssa, author.

Title: Bugs and errors with Wreck-it Ralph / Allyssa Loya.

Description: Minneapolis, MN : Lerner Publications Company, [2018] | Series: Disney coding adventures | Includes bibliographical references and index. | Audience: Ages 6–9. | Audience: Grades K to 3.

Identifiers: LCCN 2018002323 (print) | LCCN 2018014085 (ebook) | ISBN 9781541524347 (eb pdf) | ISBN 9781541524316 (lb : alk. paper) | ISBN 9781541526778 (pb : alk. paper)

Subjects: LCSH: Computer algorithms—Juvenile literature. | Debugging in computer science—Juvenile literature. | Creative activities and seat work—Juvenile literature.

Classification: LCC QA76.9.A43 (ebook) | LCC QA76.9.A43 L6748 2018 (print) | DDC 005.1—dc23

LC record available at <https://lcn.loc.gov/2018002323>

Manufactured in the United States of America
1-44519-34769-4/10/2018

Table of Contents

What Are Bugs and Errors?	4
Fix It, Felix!.	6
Ralph Crashes the Party	8
<i>Hero's Duty</i>	10
Double Stripe!	12
Building a Kart	14
Escape the Nesquik-Sand.	18
Learning to Drive.	20
Random Roster Race.	22
Diet Cola Mountain	24
Fixing <i>Sugar Rush</i>	26
Keep Coding!	28
Answer Key	30
Glossary	30
Further Information	31
Index	32



What Are Bugs and Errors?

Computers are awesome! People use them every day to do work and have fun. Did you know that computers need algorithms to work? Algorithms are groups of instructions called code. They tell computers what to do.

Algorithms don't always work. They may have mistakes called bugs. For example, an algorithm to take two steps forward may look like this:



Each arrow gives an instruction. The same algorithm with a bug looks like this:



Instead of two steps forward, the algorithm says to take one step forward and one step backward. The algorithm has a bug and must be fixed, or resolved.





Wreck-It Ralph characters get into lots of trouble during their adventures. You can use algorithms to help them! But an algorithm won't work if it has bugs. So, let's make sure your algorithms are bug-free! For some of these activities, you'll need a partner, plain paper, construction paper, scissors, and pencils.

Fix It, Felix!



Fix-It Felix fixes everything. He needs to reach the window that Ralph wrecked.

Help Felix reach the broken window and learn about bugs at the same time! Look at the building on the next page. Do you see Felix on the lower-right side? Then look at the list of algorithms below the building. Choose the algorithm that will move Felix to the broken window. If an algorithm doesn't work, it has a bug.





a.



b.



c.



Check your answers on page 30.

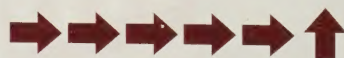
Ralph Crashes the Party



The Nicelanders are having a party. Ralph wants something to eat. Help him move across the room to the cake shaped like a building.



Look at the map on the next page. Pretend to be Ralph and run, or start, this algorithm:

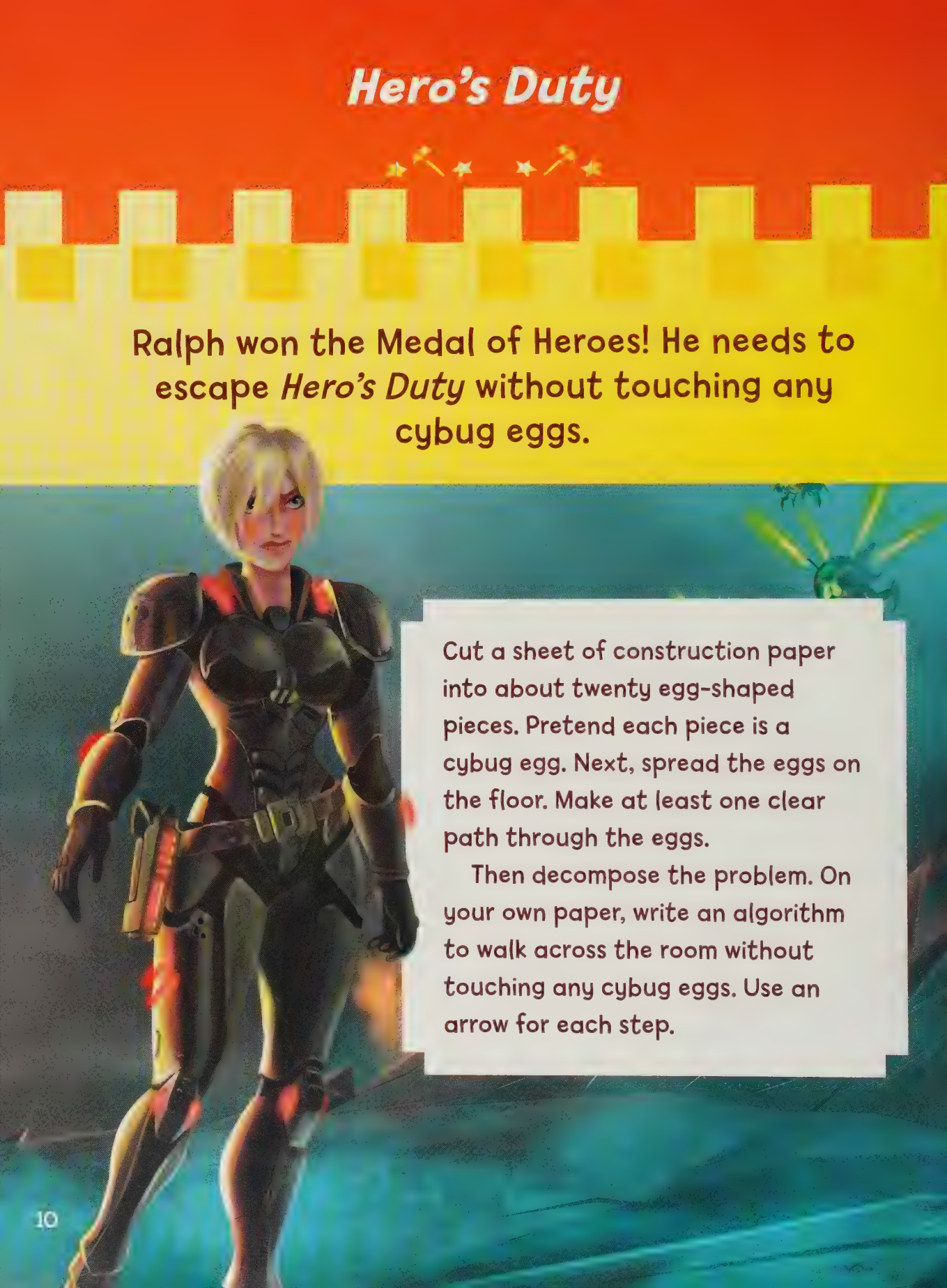




Did you bump into people? If so, the algorithm has bugs. Decompose the problem. That means think about the steps that will get Ralph to the cake. Then write a new algorithm with arrows on your own paper. Avoid bumping into people or the couch. Go to page 30 to see one possible answer.

Hero's Duty

Ralph won the Medal of Heroes! He needs to escape *Hero's Duty* without touching any cybug eggs.



Cut a sheet of construction paper into about twenty egg-shaped pieces. Pretend each piece is a cybug egg. Next, spread the eggs on the floor. Make at least one clear path through the eggs.

Then decompose the problem. On your own paper, write an algorithm to walk across the room without touching any cybug eggs. Use an arrow for each step.



Run the algorithm when you're ready. You should follow each step carefully. If you touch an egg, the algorithm has bugs. Resolve the bugs, and run the algorithm again.

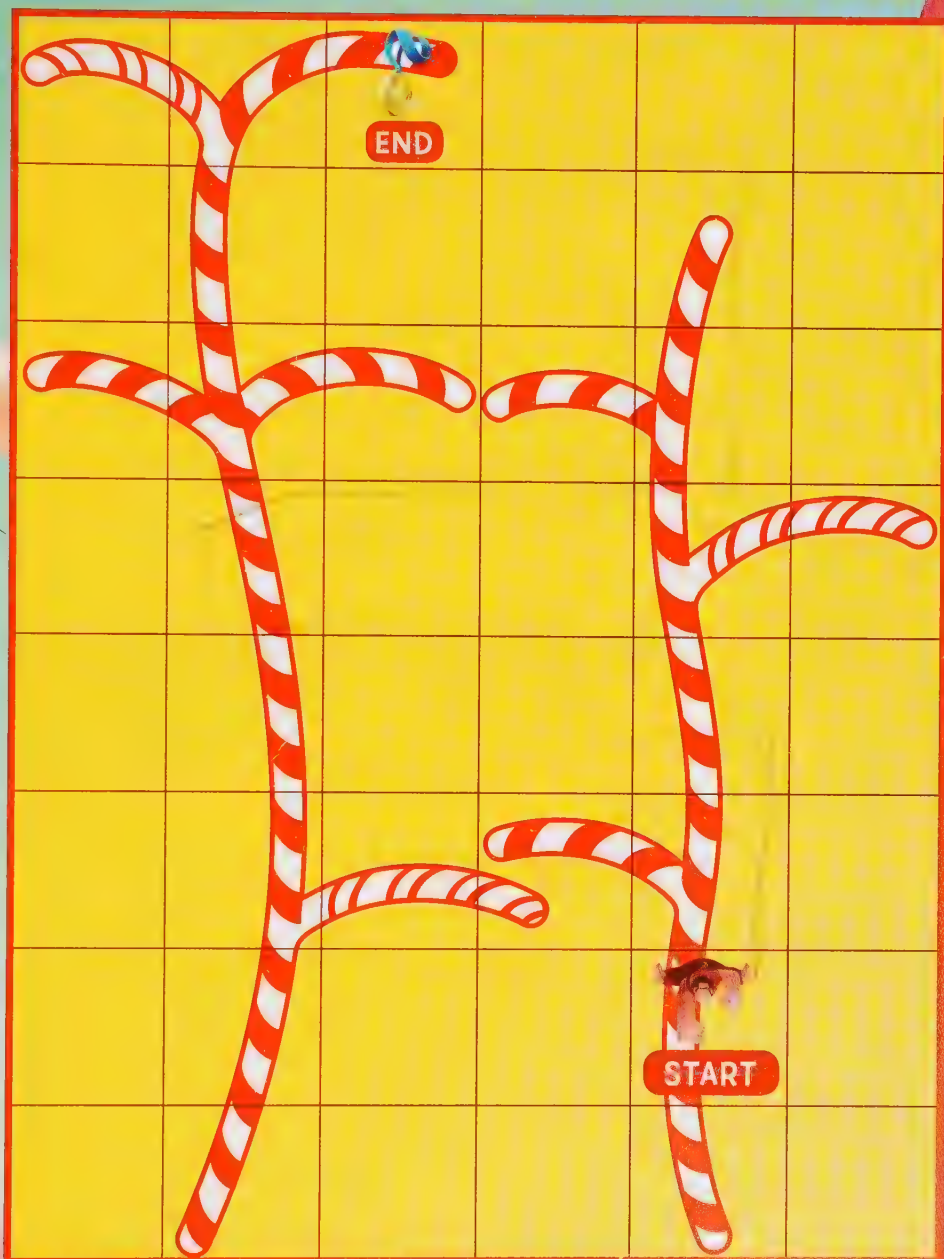
Double Stripe!

The Medal of Heroes is hanging from the top of a candy-cane tree. Ralph wants it back. Can he beat Vanellope to the top of the tree?

Ralph ran this algorithm to reach the medal:



Those steps had bugs. They brought Ralph to a branch with double stripes. Double-striped branches disappear when Ralph touches them. Look at the trees. Think about a path that avoids double stripes. Then write an algorithm on your own paper that brings Ralph to the medal.



Building a Kart

Ralph and Vanellope broke into the kart bakery. Help them build the kart of Vanellope's dreams.

Sometimes code tells a computer to repeat an action. Instead of writing the same instruction many times, you can use loops. Loops tell computers how many times to do the same thing.

Look at the grid on the next page. You could use this algorithm to move Ralph and Vanellope to the gummy worms and then the sprinkles:



Or you could use an algorithm with loops:



The numbers tell a computer how many times to repeat an action. The loops are easier to write.

START



What sweet items would you pick if you made a kart? Make a list on your own paper of three sweets from the grid on page 17. Then give your list to a partner. Your partner will write an algorithm to move Ralph and Vanellope to each sweet. Make sure your partner uses loops! Run the algorithm together when your partner is finished. Does it work? If not, it has bugs. Resolve it and try again. When you're finished, switch and have your partner make the list of sweets.





START



Escape the Nesquik-Sand

Felix and Sergeant Calhoun are stuck in Nesquik-sand! If they can make the Laffy Taffy vines laugh, Felix and Calhoun can grab the vines and escape.

Think of ten of your favorite jokes. If you don't know ten jokes, check out a jokebook from the library. Then grab a stuffed animal, and find a partner.

Face your partner from five steps away. Your partner will pretend to be Laffy Taffy. You and the stuffed animal are Calhoun and Felix. It's time to make your partner laugh!





Think of each joke as a line of code. Read a joke out loud. Did your partner laugh? If so, your partner will take one step forward. If your partner didn't laugh, the joke has a bug. Move on to the next joke. Calhoun and Felix will be saved when Laffy Taffy laughs five times. And you'll have an algorithm of funny jokes with no bugs!

Learning to Drive

Vanellope wants to race. She practices with Ralph inside Diet Cola Mountain.

Look at the paths on the next page. Vanellope can drive straight if the path is clear. But if there's a hole in the path, she must swerve to avoid it. Each hole should have a swerve symbol below it. If it doesn't, it has bugs.



Run the algorithm below each path. Did you find all the bugs?
Write new algorithms on your own paper that will get Vanellope to the end of each path.

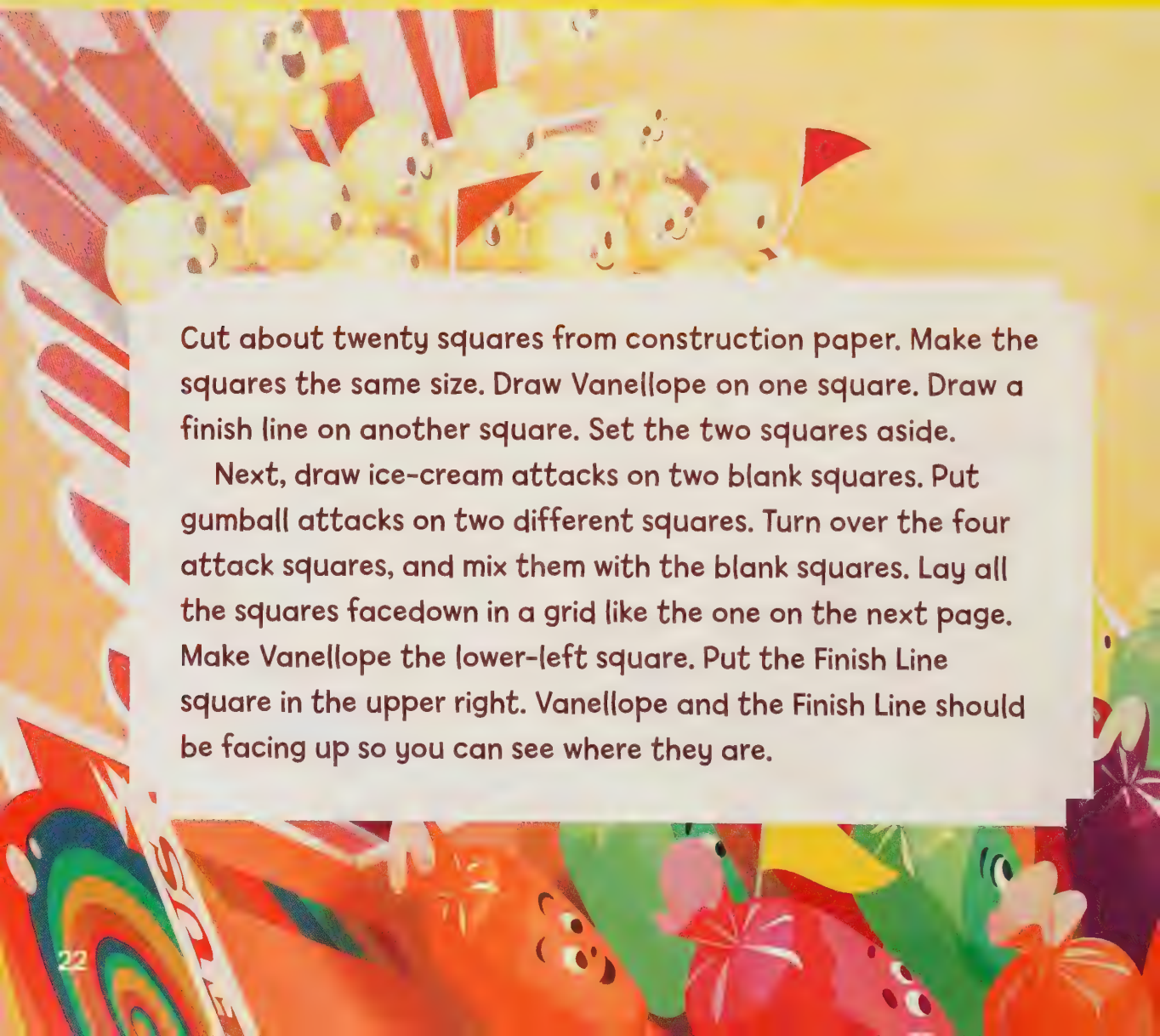
drive straight = ➡

swerve = 🌀➡



Random Roster Race

It's time for the Random Roster Race. Watch out for gumballs and ice-cream attacks!



Cut about twenty squares from construction paper. Make the squares the same size. Draw Vanellope on one square. Draw a finish line on another square. Set the two squares aside.

Next, draw ice-cream attacks on two blank squares. Put gumball attacks on two different squares. Turn over the four attack squares, and mix them with the blank squares. Lay all the squares facedown in a grid like the one on the next page. Make Vanellope the lower-left square. Put the Finish Line square in the upper right. Vanellope and the Finish Line should be facing up so you can see where they are.



Write an algorithm on plain paper to get Vanellope to the Finish Line. Remember to use looped lines of code. Run the algorithm, and turn the squares over one by one. If you run into a trap, that means your algorithm has a bug. Rewrite it to go around the bug. Keep going until your algorithm reaches the Finish Line.



Diet Cola Mountain

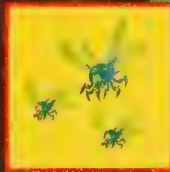
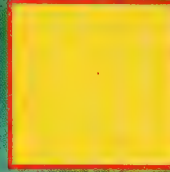
Ralph's friends are trapped by cybugs! He can save them if he smashes the top of Diet Cola Mountain. But first, he needs to get up there.

Ralph can't climb with all these cybugs in the way. He needs to push away each cybug before he can do the next step. Ralph tried to follow this algorithm to climb the mountain:

2(↑) 2(→) 3(↑) 3(push) 2(↑) → 2(push)



END



START

The algorithm has bugs, though. The steps are in the wrong order.

Look at the grid on this page. Then fix the algorithm. Change the order of the steps until it brings Ralph to the top of Diet Cola Mountain.

Fixing *Sugar Rush*

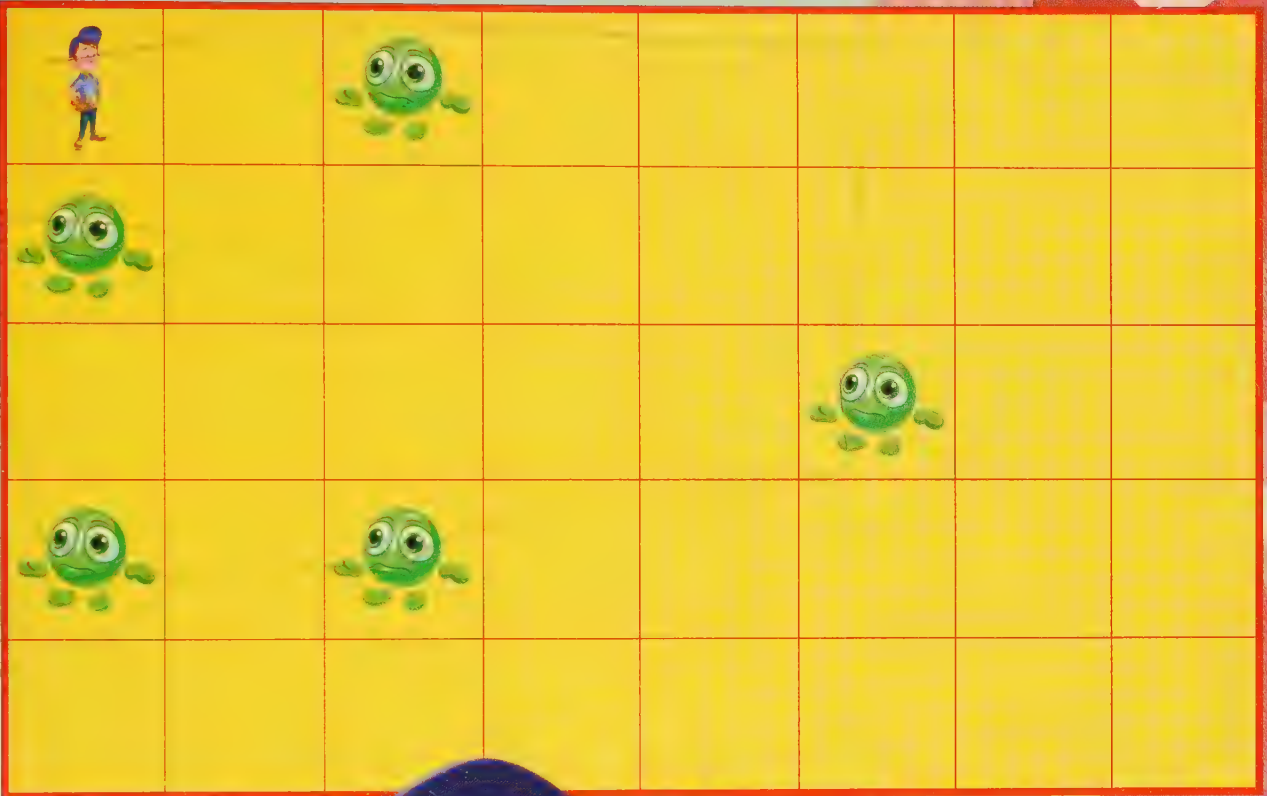
The cybugs are gone! But they left behind a big mess. *Sugar Rush* needs to be fixed.

In this project, you and your partner will each draw a grid like the one on the next page. Draw Fix-It Felix in the upper-left square. Add Sour Bill to five of the squares. These are the places where the world of *Sugar Rush* is broken, or has errors. You can put Sour Bill anywhere.

Trade grids with your partner. Write an algorithm to move Fix-It Felix to each square where Sour Bill appears on your partner's grid. Use Felix's hammer to fix each error. For example, this algorithm could fix the grid on page 27:

↓ ↗ 2(↓) ↗ 2(→) ↗ 3(↑) ↗ 3(→) 2(↓) ↗

START



Does your algorithm fix every square where Sour Bill appears on your partner's grid? If not, resolve the bugs in your algorithm and run it again.

Keep Coding!

After learning about bugs and errors, you may start seeing them everywhere. Mistakes people make are like bugs. Think about mistakes you see in your daily life.

Was the school bus late in the morning? Did you trip walking up the steps? Did your pencil break? Did your friend drop some food during lunch?



When a mistake happens, think about algorithms. What steps led to the mistake? Was your shoelace untied? Maybe you pressed too hard on your pencil. Then think about how to fix the error and avoid the mistake in the future. Soon you'll be great at spotting and fixing bugs!



Answer Key

Page 7: b

Page 9: ↑→→→→→

Page 13: ↑↑↑↑←←←↑↑→

Page 21: →↪→↪

→↪→↪↪→

↪→↪↪↪↪→↪→

Page 25: 2(↑) 2(push) 2(→) 2(↑) 3(push) → 3(↑)

Glossary

algorithm: instructions, made up of lines of code, that tell a computer how to solve a problem or finish a job

bug: a mistake in lines of code

code: instructions that computers can follow

decompose: to take a big problem and break it down into small pieces to figure it out

loop: a line of code that tells a computer to repeat an instruction a certain number of times

resolve: fix a mistake (a bug) in a line of code

run: to start an algorithm



Further Information

CodeMonkey

<https://www.playcodemonkey.com>

Code.org

<https://code.org/learn>

Kelly, James F. *The Story of Coding*. New York: DK, 2017.

Lyons, Heather, and Elizabeth Tweedale. *Coding, Bugs, and Fixes*. Minneapolis: Lerner Publications, 2017.

Prottzman, Kiki. *My First Coding Book*. New York: DK, 2017.



Index

algorithms, 4-6, 8-12, 14, 16, 19, 21, 23-27, 29
arrows, 4, 9-10
decompose, 9-10
instructions, 4, 14
loops, 14, 16, 23

About the Author

Allyssa Loya is an elementary school librarian in North Texas. Her passion for bringing meaningful learning to students led her to cultivate a technology-forward library that includes a makerspace and a coding club. While running the coding club in the library, she realized how important it is for every student to experience coding. Not every student will grow up to be a computer programmer, but all students will need to know how to think clearly and critically when they are adults.

Loya is married to an IT manager, who is a perfect support system for her technological endeavors. Her two young boys are a constant reminder of the experiences that all students deserve from their educators.



Check Back in the Book

This book shows you how coding can be used to solve problems the characters faced in the story. Think about these questions as you work your way through this activity book.

- What kinds of problems or errors do the characters come across?
- What kinds of bugs can make solving those problems difficult?
- How can algorithms help the characters fix the bugs and errors?



Get a Move On!

As you read this book, you saw examples in which code was used to move a character. There are three main goals for moving a character.

1. To Avoid or Get Around Something
2. To Get To Something and Use It
3. To Get To Something and Destroy It

NOW TRY THIS

Choose one of the these goals. Find an example in the book. Then, work with a partner and follow these steps to reach the goal.

Decide where the character starts and needs to go.

Decide why the character needs to go there.

Decide what needs to be done once the character is there.

Decide how to use coding to get the character there.

In Your Own Words

Imagine that someone who doesn't know anything about coding asks you about it. How would you answer these questions in a simple way that would help a non-coder understand?

1. What is an **algorithm**?
2. What do **bugs** have to do with computers?
3. What does it mean to **decompose a problem**?



Computers can do lots of amazing things. Do they ever make mistakes? Of course! People write code to tell computers what to do. But sometimes the code has a bug, or an error. You can help fix bugs in code! You don't even need a computer to learn about bugs and errors. Ralph, Vanellope, and other *Wreck-It Ralph* characters will help you learn about bugs and how to get rid of them.

Dive into coding concepts, including algorithms, bugs and errors, looping, and conditionals, through fun activities. No computers necessary! You can learn to think like a coder with these hands-on activities inspired by your favorite Disney characters.

